AFRL-MN-EG-TP-2006-7406

A COMPARISON OF TRAJECTORY DETERMINATION APPROACHES FOR SMALL UAVS

Johnny H. Evers
Air Force Research Laboratory
Munitions Directorate
AFRL/MNGN
Eglin AFB, FL 32542-6810

Murray B. Anderson
Juan L. Lopez
Jacobs Sverdrup - TEAS Group
Eglin AFB, FL 32542-6810

SEPTEMBER 2006

CONFERENCE PAPER

This paper was presented at the AIAA Guidance, Navigation, and Control Conference, Keystone, CO, August 21-24, 2006.

AIR FORCE RESEARCH LABORATORY, MUNITIONS DIRECTORATE

■ **Air Force Materiel Command** ■ **United States Air Force** ■ **Eglin Air Force Base**

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
XX-09-2006	Conference Paper	8/31/01 - 09/01/06

4. TITLE AND SUBTITLE	
A Comparison of Trajectory Determination Approaches for Small UAVs	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 62602F
6. AUTHOR(S) Murray B. Anderson; Juan L. Lopez; Johnny H. Evers	**5d. PROJECT NUMBER** 2304
	5e. TASK NUMBER AM
	5f. WORK UNIT NUMBER 05

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Research Laboratory, Munitions Directorate, AFRL/MNGN, Eglin AFB, FL 32542-6810; Jacobs Sverdrup - TEAS Group, Eglin AFB, FL 32542-6810	AFRL-MN-EG-TP-2006-7406
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
Air Force Research Laboratory, Munitions Directorate, AFRL/MNGN, Eglin AFB, FL 32542-6810	AFRL-MN-EG
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) Same as block 8

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION A: Approved for public release; distribution unlimited.
Approval Confirmation #AAC/PA 07-11-06-333

13. SUPPLEMENTARY NOTES

This paper was presented at the AIAA Guidance, Navigation, and Control Conference, Keystone, CO, August 21-24, 2006. One of the authors is a U.S. Government employee working within the scope of his position; therefore, the U.S. Government is joint owner of the work. (Cont. on abstract)

14. ABSTRACT

*If published, AIAA may assert copyright. If so, the U.S. Government has the right to copy, distribute, and use the work. Any other form of use is subject to copyright restrictions.

In considering the problem of small unmanned aerial vehicle (SUAV) surveillance mission in a target rich environment, it is desirable to follow a trajectory path that maximizes targets coverage and observation time, while minimizing airframe maneuvering. Motivated by this requirement, this paper investigates the merits of multiple vehicle trajectory path schemes. Genetic Algorithms (GAs) and local optimum techniques are compared to a more conventional defined-path approach. The authors also introduce a polygon boundary reflection algorithm (PBRA) and investigate its merits.

15. SUBJECT TERMS

Biomimetic Control, Biomimetic Guidance, Structure from Motion, Egomotion from Optic Flow

16. SECURITY CLASSIFICATION OF: a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR	14	Johnny Evers
					19b. TELEPHONE NUMBER *(Include area code)*

A Comparison of Trajectory Determination Approaches for Small UAVs

Dr. Murray B. Anderson[*] and Dr. Juan L. Lopez[†]
Jacobs Sverdrup/TEAS Group, Eglin Air Force Base, FL, 32542

Mr. Johnny H. Evers[‡]
Munitions Directorate, Air Force Research Laboratory, Eglin Air Force Base, FL, 32542

In considering the problem of small unmanned aerial vehicle (SUAV) surveillance mission in a target rich environment, it is desirable to follow a trajectory path that maximizes targets coverage and observation time, while minimizing airframe maneuvering. Motivated by this requirement, this paper investigates the merits of multiple vehicle trajectory path schemes. Genetic Algorithms (GAs) and local optimum techniques are compared to a more conventional defined-path approach. The authors also introduce a polygon boundary reflection algorithm (PBRA) and investigate its merits. Given a scenario containing multiple targets of unknown positions, the GA optimization approach determines the waypoints defining a path that best satisfies three goals: 1) maximize the number of targets seen, 2) maximize the average observation time for each target, and 3) minimize the SUAV acceleration history. Were the target locations known apriori, this problem could decompose into a variant of the much-studied traveling salesman problem (TSP). The complication of not knowing the actual target locations apriori means that the optimization tool must find waypoints that best satisfy the multiple objectives with little actual knowledge at initiation. Given this additional complexity and the fact that there are multiple objectives that must be maximized, a GA approach was investigated because it offers the ability to rigorously search for the optimum waypoint locations while simultaneously examining performance against multiple objectives. The GA software used in the analysis is IMPROVE© (Implicit Multi-objective Parameter Optimization via Evolution)[1]. Comparison results of the GA based approaches, pareto and non-pareto, were investigated and compared with the simple PBRA and the popular Serpentine path approach. The analysis shows the GA optimization benefits and performance tradeoffs for all the path planning approaches that were studied.

Nomenclature

$\forall$	=	for all
$\exists$	=	exists
a_m	=	acceleration of the SUAV
$\mathbf{A}_i$	=	member A's performance in goal i
$\mathbf{B}_i$	=	member B's performance in goal i
i	=	index reference for goals
t_f	=	final time
t_0	=	Initial (start) time
Δti	=	average time i^{th} object is observed

[*] Director, Air Armament Technology Support Department, 308 West D Avenue, Bldg 260, Associate Fellow.
[†] Chief Scientist, 308 West D Avenue, Bldg 260, Senior Member.
[‡] Senior Researcher, AFRL/MNG, Senior Member.

I. Introduction

The analysis of trajectories for SUAVs has recently been the subject of optimization research. As the utility of these vehicles grows, techniques need to be developed and evaluated that can offer insight into how best to employ these vehicles to accomplish their complex missions. A recent application is Area Dominance in which a UAV or a munition loiters for a relatively long time over an area and searches for potential targets. While searching the UAV must also avoid obstacles in its path and be capable of responding to a dynamic environment. The path planning objective is to determine the most efficient trajectory for maximum efficiency/utility. Path planning optimization and obstacle avoidance have been areas of research using a variety of techniques based on optimal control. This interest is motivated by the successful application of UAVs in recent conflicts and the diversified role that UAVs can take in future conflicts. Receding time horizon (RTH) [2-3] has become popular for control of UAVs, particularly in determining the path for obstacle avoidance. RTH is advantageous because the computational resources needed are low and it can respond to a dynamic environment. Optimization of the complex cost function using mixed integer linear programming was discussed for the obstacle avoidance problem, including target estimation uncertainties in the model[4]. Investigations using a receding horizon control strategy to enable a UAV to fly autonomously in a complex urban environment have been presented providing a path planning approach by selecting a series of locally–optimal waypoints ahead of the vehicle[5]. Dynamic programming[6], iterative search methods [7], and random methods based on genetic algorithms[8] (GAs) have been investigated.

This research paper compares different methods to develop navigation waypoints that yield optimal performance in three primary optimization goals: 1) maximize the number of targets seen, 2) maximize the average observation time for each target, and 3) minimize the SUAV acceleration history for endurance. GA results are compared to simpler and more popular methods for the specified cost function. For this research the GA code IMPROVE©[1], originally developed to support air data estimation[9], was used. This code has recently been used for aerodynamic data extraction and airframe design optimization. Inputs to the GA software are user-defined parameter bounds (waypoints) and their resolution, as well as a description of the cost function (the three goals). For this application a constraint was placed on the maximum SUAV acceleration (2G's) so the scenario would be representative of a typical SUAV maneuver capability. This constraint, along with the minimization goal for the acceleration history, was implemented as a means to conserve fuel.

The GA was run in two primary modes: non-pareto and pareto using the same three goals. Searches using a pareto optimality approach operate on multiple goals/objectives simultaneously, and "winners" must outperform their competitors in all goal areas in order to survive the selection process. This rigid selection process typically means that a pareto-GA will converge less quickly than a non-pareto-GA, so one of the goals of this research was to run pareto versus non-pareto algorithms for a fixed number of generations (200) to determine which approach might be best suited for path planning.

It is assumed that a passive imaging sensor of specified field-of-view (FOV) and detection range gathers imagery while the SUAV cruises the GA-determined path within a specified time horizon (2400 seconds). The GA-directed search metrics are histograms of targets observed and the average time observing each specific target. The results using the GA-determined path are compared to a simpler, easy to implement algorithm, the PBRA approach. The PBRA consists of defining a polygon-shaped area containing the N-targets, and performing random path change reflection at the boundaries within the airframe maneuver constraint. Basically, the SUAV flies straight until approaching a boundary, then changes course in a random fashion within limits, then continues flying straight until the next polygon boundary is approached, and the process continues. The PBRA and the GA models are both capable of including exclusion zones inside the search area. For simplicity, a constant velocity SUAV is assumed, however the algorithm could also control the throttle within specified speed limits to improve the performance metrics. Results of simulations are displayed showing the performance of the different techniques using a synthetically generated environment. The popular Serpentine Search approach is also investigated for the given target scenario. A single SUAV is assumed; however, data link communication in a formation of UAVs is feasible and this capability could improve path planning and enhance target search and detect. This is a field of future research for path planning with the numerous vehicles exchanging position and path information.

II. Genetic Algorithms

Genetic algorithms are encompassed within the broader computer science field known as artificial intelligence. Genetic algorithms are so called because they attempt to use the supposition of evolution as a basic mechanism for improvement (i.e. learning/survival-of-the-fittest) in solving a problem. All genetic algorithm work stems from the pioneering efforts of John Holland[10], whose classic book "Adaptation in Natural and Artificial Systems" set the foundation for population-based adaptive optimizers. Following the terminology of true genetics researchers, the computational genetic algorithms developed by Holland (and his students) encode potential solutions into chromosome-like structures, then allowing these structures to compete, reproduce, and mutate to produce (hopefully) better and better solutions over time. GAs have been increasingly used in optimization studies over the past decade, and have more recently been used in multi-disciplinary optimization. There should be an emphasis on improvement rather than optimization in a multi-disciplinary context, simply because for complex problems it is not possible to prove that you have reached an optimum. In the preface to the latest reprint of his book, Holland himself says that "about the only change I would make would be to put more emphasis on improvement and less on optimization." Certainly no analytic determination of an optimum can be made for a sufficiently non-linear multi-variable problem, so consistent with Holland's observations, the emphasis in this research is on using GAs as an improvement tool.

In a strict biological sense, genetic algorithms cannot correctly be called an evolutionary model, though it is quite popular in literature to call them evolutionary because the algorithm attempts to adapt its pre-existing genes to perform best in the current environment. The algorithm does not create new genes if the current ones do not suffice, and hence the supposition of evolution does not strictly apply to the genetic algorithm approach invented by Holland and used in this research. Adaptation more correctly describes the process by which genetic algorithms find good solutions to their environment.

Many facets control the way that a genetic algorithm works. A potential solution first has to be encoded, along with all the other potential solutions that form a generation. This population is then fed one at a time to the objective function so that a measure of the performance of each member of the population can be ascertained. The better performers then have a higher probability of surviving to reproduce the next generation. Mating between survivors is the mechanism by which new populations are formed. Mutation is also allowed to occur and helps preserve genetic diversity. Over time, as generations build upon the successes of previous generations, the performance of the entire population increases as the algorithm "learns" what allele values produce good answers. Poor performers die off (lower reproduction rate) and over many generations the best performers within a given population will hopefully provide suitable performance through the objective function. Many researchers tout genetic algorithms as "global" optimizers, and that is true because of mutation and the general probabilistic non-gradient nature of GAs, but in engineering applications you typically want good answers (answers that will suffice) as fast as you can get them. Whether the "answer" is the absolute global optimum or not is less of a concern than whether you were able to get a good answer in the time allotted by management circumstances. Besides, for complicated problems there is no analytical way to determine the global optimum, so arguments over whether you have the true optimum are academic.

Pareto GAs[11-13] differ from generational GAs in that they operate on multiple goals or objectives simultaneously rather than having one "fitness" function. The goals are optimized individually to determine parameter sets that work well for each goal. Through the mating of good parameter sets for each individual goal, a family of parameter sets which work well across the spectrum of goals is obtained. This family of solutions is called the pareto optimal (p-optimal) set for the multi-objective/multi-goal problem. The operation of the pareto scheme in the genetic algorithm software is by domination. One goal set, defined as the collection of individual goal performances based on one parameter set, must clearly dominate another if the parameter set is to survive. Goal set **A** dominates set **B** if the following two statements are true:

$$\forall_i (\mathbf{A_i} \geq \mathbf{B_i}) \quad \text{and} \quad \exists_i (\mathbf{A}_i > \mathbf{B}_i)$$

When two members of the population are chosen for the tournament selection procedure, the domination rules are examined to see whether one member dominates the other. The clear winner is retained for the next generation. If there is no clear winner, a situation called nondomination, the winner is selected at random for survival.

One of the first successful implementations of a pareto strategy was performed by Schaffer. In his dissertation, Schaffer formed subpopulations for each goal area. The members of the subpopulations would compete for survival only within the subpopulation, but mating was allowed between subpopulations. One of the weaknesses of this approach, which was noted by Schaffer and modified in his later work[14,15], was that the subpopulations would essentially form niches of high fitness, but there would be few members of the subpopulations that work well in all the goal areas. Producing niches of high fitness is counterproductive in true multi-objective optimization. The IMPROVE© code uses a single population, minimizing the chance of niches. But if niches are desired, the IMPROVE© code will allow niches to form if elitism is selected in conjunction with the pareto algorithm. Elitism will preserve the best performer(s) in each niche.

III. Problem Setup

For simplicity, the SUAV is modeled as a 2-DOF point mass with a constant velocity of 100 ft/sec and a maximum turning capability of 2G's. The seeker is assumed to have a viewing range of 3000 feet with a ± 15 degree FOV. The target area is a 20,000 foot by 20,000 foot square, with 20 randomly placed targets (the same configuration was used for each case – see Figure 1). The SUAV had 25 navigation waypoints to fly through, though the methodology to fly through the points varied depending upon the case investigated. As the SUAV flies through the waypoints, statistics such as target number and time within the seeker FOV are captured. As the SUAV turns, the integral of the acceleration squared is also calculated to provide information about the energy expended during all turns. The total time of flight was set at 2400 seconds (40 minutes), so once each waypoint has been reached the vehicle can begin revisiting the waypoints in accordance with the particular algorithm being employed to navigate through the waypoints. It is understood that the results of this research cannot be applied exactly to another unique target set since the waypoints determined in this study only apply to the 20 targets shown in Figure 1, but that was not the goal of this research. Rather, the investigation of possible trajectory optimization schemes using a GA was the focus of the research. Another goal was to see what "features" could be extracted from the resulting trajectories that might help guide future research. Since the GA has no preconceived ideas of what a trajectory should look like, any inherent bias on the part of the authors is removed from the problem. This unbiased approach to examining problems makes GAs extremely useful for optimization studies, and this is what makes GAs so unique. Unlike other optimization approaches, a GA does not start out with an initial "guess" or starting solution.

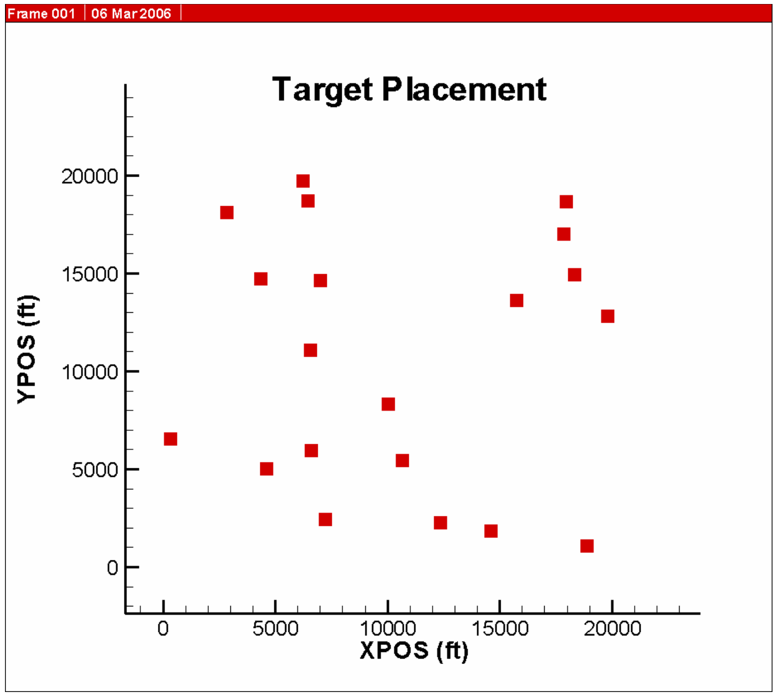

Figure 1. Target Placement for All Runs

Optimization Cost Function:

As described in the Introduction, the multi-objective optimization goals for the GA are: 1) maximize number of targets seen, $Nt = \sum_{i=1}^{N} Ni$; 2) maximize the average observation time for each object: Δti, while 3) minimizing the value of acceleration squared over the total time of flight between waypoints: $\min \int_{t_0}^{t_f} a_m{}^2 dt$

Case 1: Pareto GA, GA-Determined Waypoints and Path

For this run the GA had 50 independent variables to define (the 25 waypoints x and y coordinates of each waypoint). The coordinates were numbered 1-25 and the vehicle flew these waypoints in order, so depending upon the values of each x and y coordinate, the GA had the flexibility of moving these waypoints around to lengthen or shorten particular legs, or, through the location of these waypoints since they are flown in order, to determine the best direction of travel through particular portions of the target area. This is an important distinction that separates this optimization problem from the standard traveling salesman problem. In the TSP, the usual optimization goal is to minimize distance traveled while visiting each waypoint. For the current research, the direction of travel during each leg can the most important parameter in determining whether a target is seen and for how long. When the

SUAV has visited all waypoints, it returns to waypoint 1 and begins revisiting the waypoints, eventually terminating when the maximum time of flight has been reached.

Each x and y coordinate could vary between 0 and 20,000 feet, with a resolution of 200 feet. Each x and y coordinate, therefore, requires 7 bits for a total of 350 bits for the entire problem. These 350 bits represent 2^{350} (2.29×10^{105}) possible solutions using a GA with a binary alphabet representation. The pareto GA treats each goal as separate and of equal importance. During tournament selection, only solutions that are clearly better in each goal are guaranteed to win the tournament. For this run, an elitism strategy with creep mutation at a 5% rate was used in conjunction with a 90% crossover rate and 0.1% random mutation rate. The population size was set to 1000 members to ensure good genetic diversity, and the maximum number of generations was set to 200. These 200,000 trajectories required approximately 2 ½ days to run on a 1.7GHz Windows XP computer. All other GA runs in this research used the same population size, creep rate, crossover rate, and mutation rate. Each GA run required the same computer run time.

Case 2: Non-Pareto GA, GA-Determined Waypoints and Path

For this case, automatic fitness scaling was used on each goal and the fitness scaling was based upon the "best" output in each goal. For example, if the "best" performer for a generation saw the targets for an average of 15 seconds, then each member of the population would be scaled so that performance in this goal is normalized and ranged between 0 and 1. The same process was used for the other two goals, and then the sum of these three normalized goals was used in the tournament selection process to determine the "winners". As before, the GA setup was identical to Case 1 except for the fact that this was a non-pareto run. The GA still had to determine performance in three goal areas based upon 50 input variables, and still had to determine the optimal location of waypoints and the resulting path when flying through the waypoints 1-25 in that order.

Case 3: Non-Pareto GA, GA-Determined Waypoints, Waypoint Visitation Based upon Nearest Neighbor

For this case, the GA setup was identical to Case 2; however, the path determination algorithm was changed so that the waypoints were visited based upon "nearest neighbor." In this case, the GA was not responsible for optimizing the path, but merely the waypoints. The objective of this case was to determine the relative significance of allowing the GA to determine the path as was done in Cases 1 and 2. The "nearest neighbor" approach is very simple. First, the vehicle calculates the nearest unvisited waypoint and heads toward that waypoint. Once that waypoint is reached, the waypoint is logged as "visited" and the SUAV heads toward the next nearest unvisited waypoint. This process continues until all waypoints have been visited. When all the waypoints have been visited, the algorithm resets all the waypoints as "unvisited" and begins again. Termination occurs when the maximum time of flight has been reached.

Case 4: Comparison Case, Non-GA, Standard Polygon Boundary Reflection Algorithm (PBRA)

The PBRA is a simple approach to searching a target area. The SUAV is started at a coordinates (0,0) with a random heading. The vehicle flies along a straight line until it reaches a boundary defined by the 20,000 foot by 20,000 foot square search area. The angle of incidence to this boundary, coupled with a random amount of variation so that the reflection is not an exact mirror of the angle of incidence, is used to determine the reflection angle that will send the SUAV back into the target area. The algorithm continues until the maximum time of flight is reached. Statistics are kept as in the other cases, documenting which targets are seen and for how long. The PBRA ran in seconds on a 1.7 GHz Windows XP computer.

Case 5: Comparison Case, Non-GA, Point-to-Point Local Optimum Search with Fixed Length Legs

This second comparison case uses fixed time-of-flight heading variations from a starting point of (0,0) with a heading of 0 degrees. Each of the 25 flight legs is 96 seconds long for a total time of flight of 2400 seconds as in the other cases. From the starting point, the "local optimum" heading is found by allowing the SUAV to look along flight paths +/- 180 degrees from it's current heading and position. The heading, and resulting path, that produces the most time spent looking at targets while minimizing the required turn from the current heading is the "local optimum" heading, and the vehicle flies along this heading until it reaches the end of this flight leg. At the end of the flight leg, another look +/- 180 degrees will yield another "local optimum" heading, but this time the targets must be targets that have not been seen before. This process (fly-look-fly) is continued until all targets have been seen, whereupon the process begins anew. The fly-look-fly process is continued until all 25 legs of the mission have been flown. As with the other cases, statistics in all three goal areas are captured for comparison purposes. The fly-look-fly process has an inherent advantage over the other methods, namely, the optimum headings are derived from knowing what targets will be seen and for how long if a particular heading is selected. This particular algorithm took several minutes to run on a 1.7 GHz Windows XP computer.

Case 6: Comparison Case, Non-GA, Point-to-Point Local Optimum Search with Variable Length Legs

This comparison case is very similar to Case 5, but in this case the length of the legs is determined through iteration. The algorithm, for all flight paths +/- 180 degrees from its current heading and position, increments time flown in any given direction to determine whether it would be beneficial to lengthen each potential leg. Benefit, in this case, is determined by whether time spent looking at targets increases the longer the SUAV flies in a particular direction. When the algorithm determines that there is no additional benefit to continue on the current heading, the leg terminates and the algorithm begins searching for the next direction/leg-length to fly. Since some legs will likely be shorter than the prescribed 96 seconds used in Case 5, the number of legs is not specified apriori. Rather, flight is simply terminated at 2400 seconds. This particular algorithm took several minutes to run on a 1.7 GHz Windows XP computer.

Case 7: Comparison Case, Non-GA, Conventional Serpentine Path Search

This case was included because it is very typical of the types of search path algorithms that have been used over the years. The algorithm is very simple; the SUAV starts at the south end of the search area and flies an appropriate time (~172 seconds for this target area) to traverse the search area, then turns 180 degrees and flies back across the search area, one seeker FOV width (no overlap) north from the previous scan area. This process is continued for 2400 seconds. This particular algorithm took only a few seconds to run on a 1.7 GHz Windows XP computer.

IV. Results

Case 1: Pareto GA, GA-Determined Waypoints and Path

Case 1, as a pareto GA case, has multiple solutions that map a pareto-optimal front as the solution progresses. For visualization purposes, Figure 2 shows a few selected generations (0, 100, and 200). Each member of the population has its own unique performance in each goal, and by generation 200 the survivors have established a robust set of possible solutions from which to choose. From generation 100 to generation 200, it is obvious that the GA found consistently better solutions in terms of number of targets seen and the average time those targets were seen, though at the expense of having to increase the amount of turning required to do so. In generation 100, nearly half the population saw 18 or fewer targets for less than 30 seconds. By generation 200, two-thirds of the population saw all 20 targets for more than 30 seconds. For illustration purposes, one of these generation 200 trajectories will be examined in more detail.

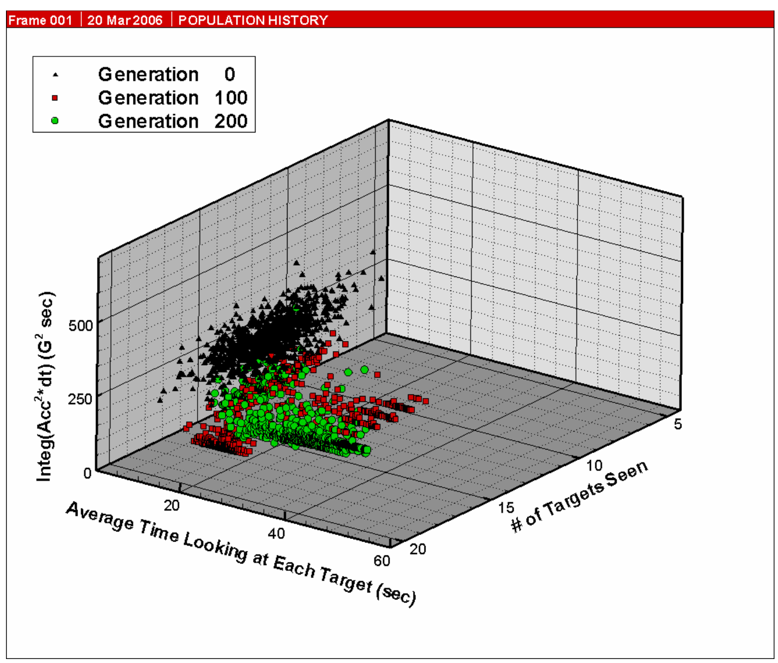

Figure 2. Pareto GA Convergence

Figure 3 shows the trajectory for member 103. The dark/wide line is the vehicle path and the light/thin lines represent the field of view seen by the seeker. As this figure shows, each target was definitely seen for some period of time, and the trajectory contains many sharp turns to enable repeated re-looks of some of the targets. The 20 targets were seen an average of 52.77 seconds with an acceleration goal value of 280.4934 G^2·sec. The entire time of flight was used to fly to each of the 25 waypoints, so no waypoints were revisited.

Many other similar trajectories were produced by other members of population 200. For example, member 352 saw all 20 targets an average of 48.2882 seconds with an acceleration goal value of 249.076 G^2·sec – a 11% energy savings over Member 103 but at a sacrifice of roughly 4.5 seconds of average target viewing time. The pareto GA gives the user a wide variety of solutions with various performance in each goal so that the user can determine which solution is more desirable.

Figure 4 shows the statistical breakdown of the amount of time each target was seen. Most targets were seen more than 30 seconds, but there were a couple of targets that were seen less than 1 second, namely targets 8 and 19. In general though, this algorithm produced good results and the GA-determined waypoints provided good coverage of the majority of the targets.

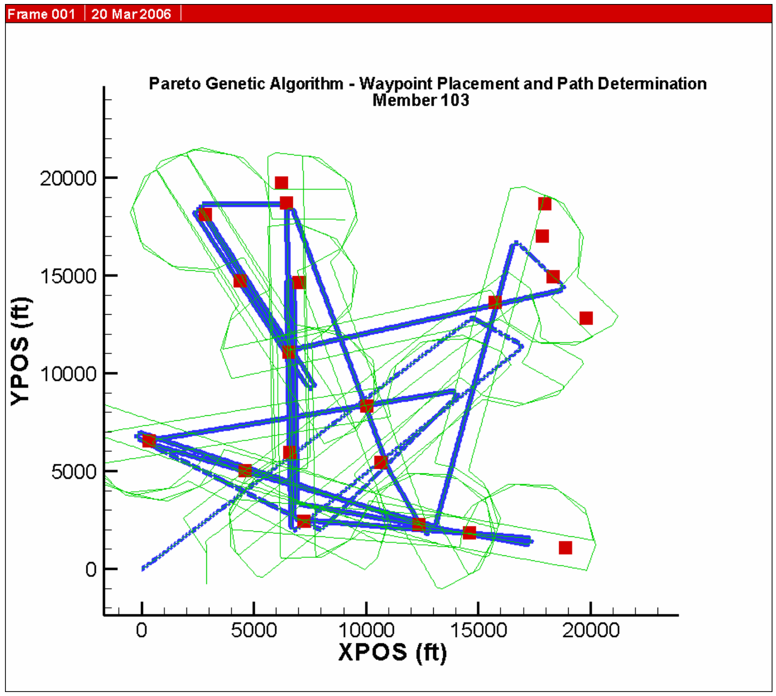

Figure 3. Pareto GA – Trajectory for Member 103

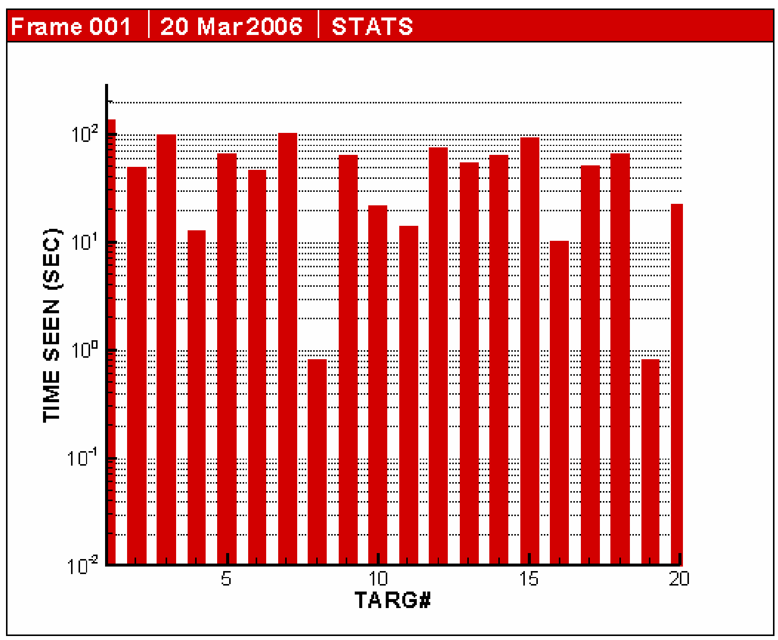

Figure 4. Amount of Time Spent on Each Target – Pareto GA, Member 103

Case 2: Non-Pareto GA, GA-Determined Waypoints and Path

The non-pareto GA run determines a "best" member of the population by measuring its performance against every other member in each goal simultaneously through auto-scaling of each goal. Figure 5 shows the trajectory of the "best" member from generation 200. As this figure shows, the GA-determined waypoints and path provided full coverage of all 20 targets. These 20 targets were seen an average of 53.357 seconds with an acceleration goal value of 175.1778 G^2·sec. This represents a savings of over 40% on the acceleration goal while at the same time increasing the average viewing time of the targets by slight more than 1 second. The pareto-GA has a very strict selection process (i.e. competing solutions must be better in all three goal areas simultaneous in order to win the selection process), and this process definitely slows convergence of the algorithm. The non-pareto GA is not encumbered by this approach, so within 200 generations the non-pareto GA has a distinct advantage.

Another interesting point is that unlike the pareto-GA case, the SUAV in this non-pareto case revisits the first three waypoints within the 2400 second time of flight, indicating an efficiency not found by the pareto GA.

Figure 6 shows the statistical breakdown of the amount of time each target was seen. Most targets were seen more than 30 seconds, and no target was seen less than 10 seconds. This contrasts nicely with the pareto-GA case, where two targets were seen less than 1 second. In general, this algorithm produced excellent results and the GA determined waypoints/path provided excellent coverage of the targets while keeping the acceleration goal at a minimum.

Case 3: Non-Pareto GA, GA-Determined Waypoints, Waypoint Visitation Based upon Nearest Neighbor

The Nearest Neighbor algorithm produced the trajectory shown in Figure 7. The dark/wide line is the vehicle path and the light/thin lines represent the field of view seen by the seeker. As this figure shows, each target was definitely seen for some period of time, and the trajectory shows the obvious point-to-point flight dictated by the Nearest Neighbor algorithm. One thing that is not so obvious is that the SUAV actually flew through each waypoint nearly three times within the 2400 second time of flight. This inherent efficiency within the point-to-point algorithm helped boost the average time each target was seen by a nearly a factor of three. In terms of performance in the three goals, the SUAV saw 20 out of 20 targets an average of 63.447 seconds. The acceleration goal had a value of 253.8274 G^2·sec. The average time spent looking at the targets was better than either of the GA-determined path cases (1 & 2), but the acceleration goal was 45% higher than for the non-pareto GA case.

Figure 8 shows the statistical breakdown of the amount of time each target was seen. Most targets were seen more than 50 seconds, but there were a couple of targets that were seen less than 10 seconds, namely targets 15 and 18. In general though, this algorithm produced excellent results and the GA-determined waypoints provided very good coverage of the targets.

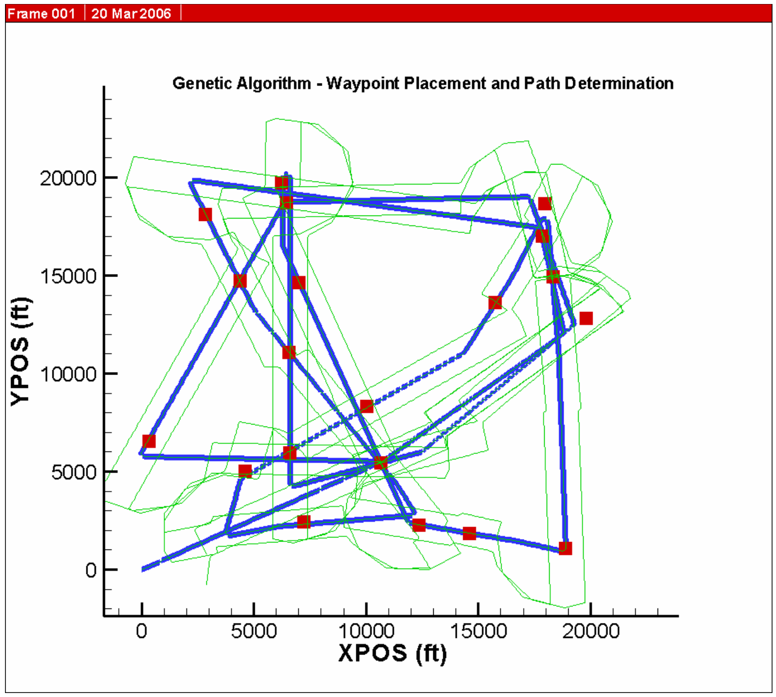

Figure 5. Non-Pareto GA, "Best" Trajectory

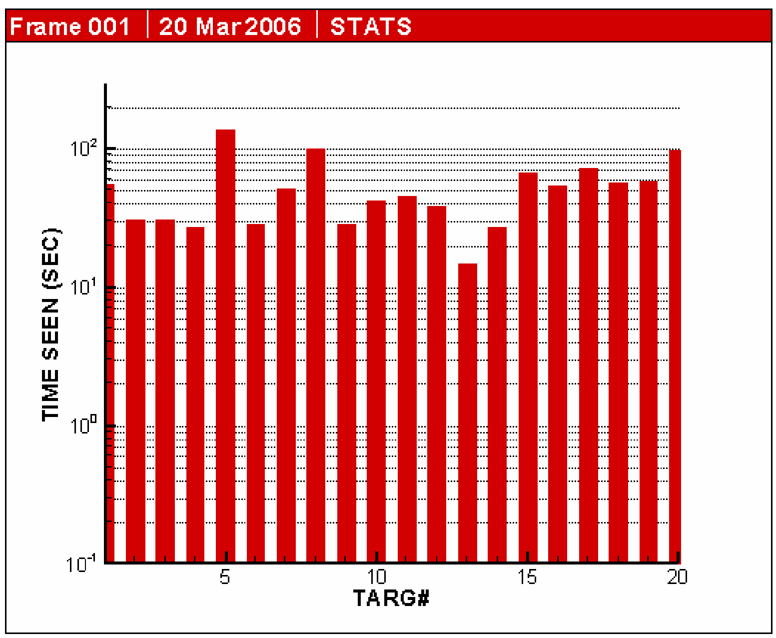

Figure 6. Amount of Time Spent on Each Target – Non-Pareto GA

Case 4: Comparison Case, Non-GA, Standard Polygon Boundary Reflection Algorithm (PBRA)

A sample PBRA trajectory is shown in Figure 9. The SUAV in this case saw 15 out of the 20 targets an average of 12.936 seconds. The acceleration goal had a value of 160.9569 G^2·sec. Ten random starts of this algorithm produced an average of 15.07 targets seen an average of 15.05 seconds with an acceleration goal of 179.492 G^2·sec. This algorithm did not produce comparable results to the any of the GA-determined waypoint approaches, but it was not expected that it would. The beauty of the PBRA, however, is that it does not require any information beyond the boundary definition and provides a degree of randomness in the trajectory that could aid in survivability.

Case 5: Comparison Case, Non-GA, Point-to-Point Local Optimum Search

As expected, the local optimum solution produces a trajectory that sees every target and minimizes the number of large-angle turns (see Figure 10). The SUAV in this case saw 20 out of the 20 targets an average of 18.981 seconds. The acceleration goal had a value of 194.3620 G^2·sec. The local optimum solution really shows that the best local optimum does not compare very favorably with any GA-optimized result. The average time each target was seen was a full 25 seconds less than the worst GA-derived solution.

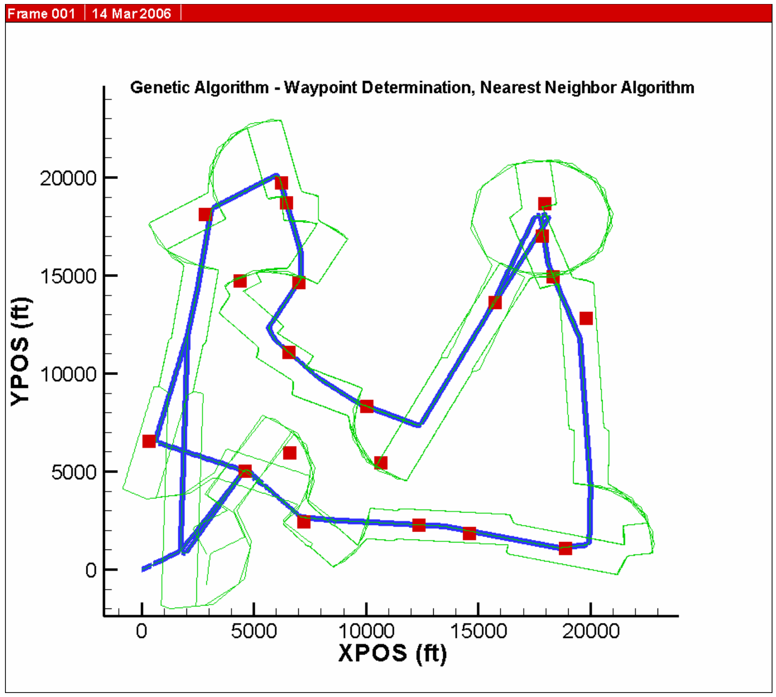

Figure 7. Nearest Neighbor Algorithm Trajectory

Case 6: Comparison Case, Non-GA, Point-to-Point Local Optimum Search with Variable Length Legs

Figure 11 shows the trajectory for this case. As expected, 20 out of 20 targets were seen, but the variable length legs approach increases the average time on target to 38.47 seconds. This is a substantial improvement over using fixed-length legs. The acceleration goal had a value of 215.8509 G^2·sec, an 11% increase from the fixed-length leg case, but the SUAV flew 42 legs versus the 25 it flew for the fixed-length leg case.

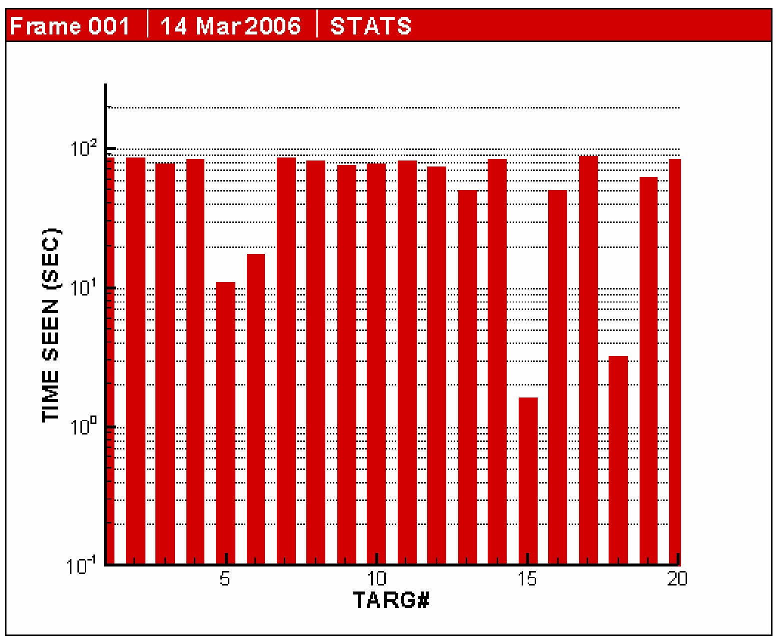

Figure 8. Amount of Time Spent on Each Target – Nearest Neighbor Algorithm

Case 7: Comparison Case, Non-GA, Conventional Serpentine Path Search

Figure 12 shows the trajectory for the serpentine search path. Only 19 of the 20 targets were seen, and these targets were seen only 9.49 seconds on-average. The target not seen was missed because of its placement near one of the turns. This conventional approach is substantially poorer than any of the GA or local optimum cases, but compares favorably with the PBRA. As expected, from an energy perspective, the serpentine path is very efficient, requiring the use of only 50.44 G^2·sec of turning energy.

V. Performance Summary

The following table summarizes the results for each approach studied in this research. It should be noted that these results apply to the selected target distribution only. For Case 1, the pareto GA, the range of values indicates the performance of the better population members from the pareto-optimal front. As the acceleration goal shows, the most efficient approach in terms of energy was the serpentine path. This result is not surprising since the turns required with this approach are fewer than any other approach (14 versus at least 25). The second most efficient approach was the non-pareto GA where the GA determined both the waypoints and the path. This result is somewhat surprising because it was expected that the PBRA would be more efficient than any of the GA-based algorithms. The PBRA algorithm inherently limits the number of turns and purposely flies a straight path until it absolutely has to turn due to the boundary. The fact that one of the GA runs found a more energy efficient solution was encouraging. In terms of average viewing time of each target, run 3, the Nearest Neighbor algorithm with GA-determined waypoints, produced the highest average viewing times. It must be remembered that the waypoints were revisited nearly 3 times for this case within the 2400 second time of flight, and even with the multiple revisits, two of the targets were seen less than 10 seconds. The non-pareto GA (Case 2), actually had better performance in terms of having all targets seen more than 10 seconds each. The non-pareto GA and the Nearest Neighbor GA both exhibit one similar feature in the trajectories, namely, they have fewer sharp turns (>90 deg) than the basic pareto GA approach. The number of turns is actually more in the Nearest Neighbor GA case than in the pareto GA, yet the overall energy used in turning is comparable. This could be a worthwhile feature that could be extracted from this research and used in future studies. The local optimum solution with fixed-length legs (Case 5), though fairly energy efficient, did not produce an average time on each target that was comparable to any of the GA solutions. Case 6, the local optimum with variable-length legs substantially improved the average time-on-target with only an 11% increase in energy requirements. This feature could be worth further

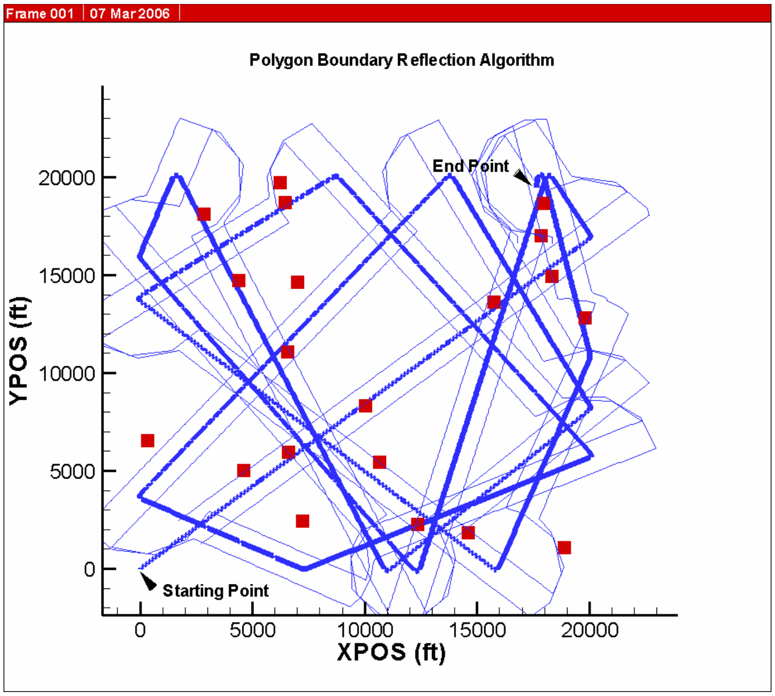

Figure 9. PBRA Trajectory

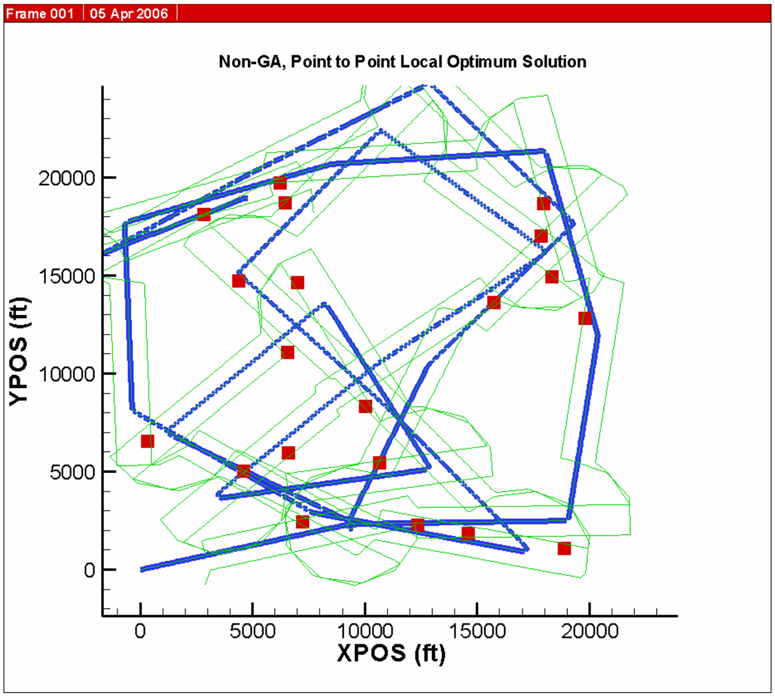

Figure 10. Local Optimum Trajectory, Fixed Length Legs

exploitation and research. The serpentine path (Case 7) trajectory, while extremely energy efficient, did not see all the targets and had very poor performance in terms of average viewing time on those targets that it did see. As all these runs show, there are many ways to increase the average viewing time of targets over the conventional serpentine path. The PBRA algorithm saw the targets an average of 5.5 seconds (58%) longer than the serpentine path. Since the average time on target was based on all 20 targets instead of just "seen" targets, the targets seen by the PBRA were seen a considerably longer time (~20 seconds on average) than the 15.05 seconds shown in Table 1. These results for the PBRA are encouraging given its simplicity and the likelihood that it will help enhance vehicle survivability because of the somewhat random path it takes through the target area.

VI. Conclusion

Both the Nearest Neighbor GA approach and the non-pareto GA approach worked well, and should be further examined through research. Also, the local optimum solution with variable-length legs was better than expected and should be further examined. A common trajectory feature of these more successful runs was the low number of sharp turns. This feature will be explored further in future research. These algorithms should also be examined against moving targets to see how these approaches compare for a more difficult target scenario. In terms of problem set-up, the average time goal could also be reconstructed so that each target must be seen a minimum amount of time, or perhaps a variance could be measured to balance the time more evenly across the targets.

Acknowledgments

This work was sponsored by the Air Force Research Laboratory, Munitions Directorate, under contract F08635-01-C-0002.

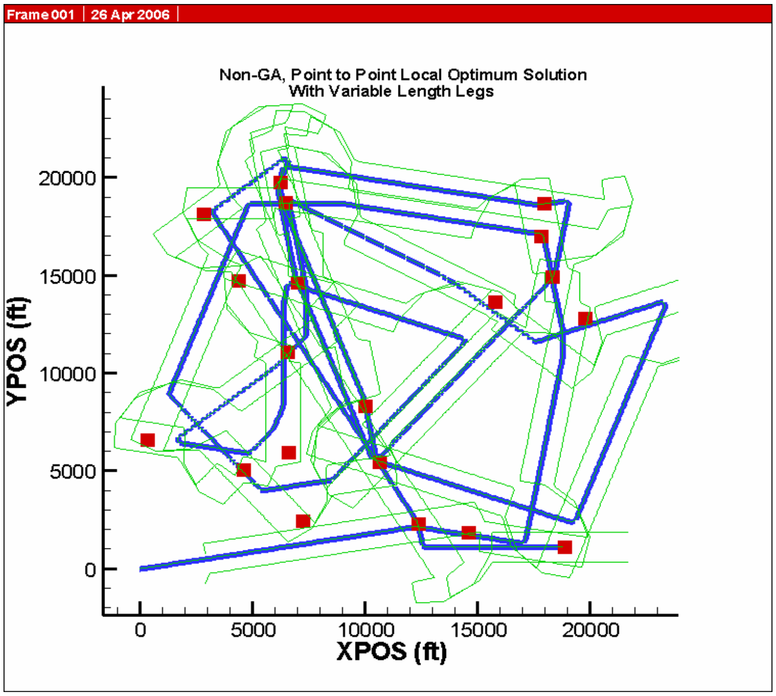

Figure 11. Local Optimum Trajectory, Variable Length Legs

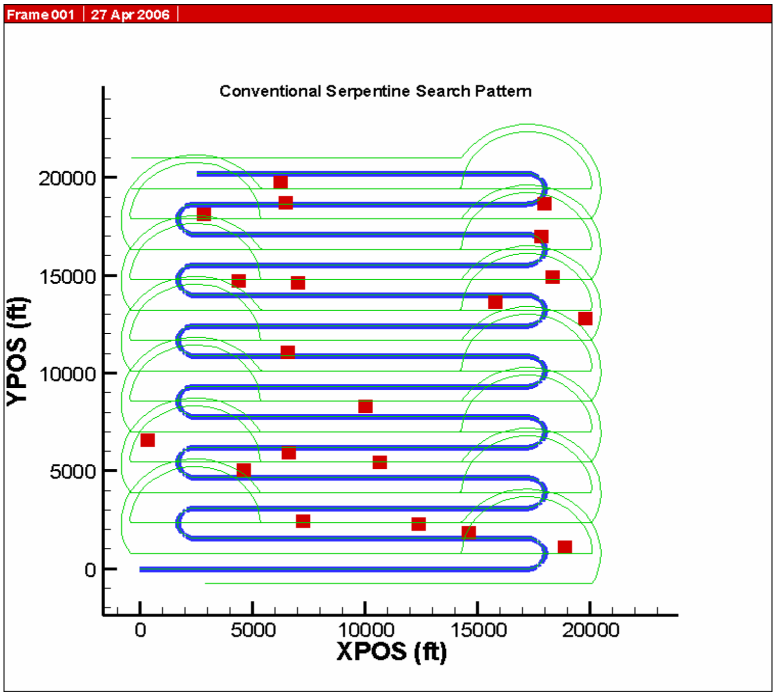

Figure 12. Serpentine Search Trajectory

Table 1. Performance Comparison of Different Algorithms/Approaches

Run	Number of Unique Targets Seen	Average Time Each Target Seen (sec)	Acceleration Goal (G^2·sec)
Case 1: Pareto, GA Determined Waypoints and Path	20	45 - 52.77	230 - 280
Case 2: Non-Pareto, GA Determined Waypoints and Path	20	53.36	175.18
Case 3: Non-Pareto, GA Determined Waypoints, Nearest Neighbor	20	63.48	253.83
Case 4: PBRA (10 random runs)	15.07	15.05	179.49
Case 5: Non-GA, Local Optimum with Fixed-Length Legs	20	18.98	194.36
Case 6: Non-GA, Local Optimum with Variable-Length Legs	20	38.47	215.85
Case 7: Serpentine Path	19	9.49	50.44

References

[1] Anderson, M.B., "Users Manual for IMPROVE Version 3.7", Copyright© 2006
[2] Kuwata,Y., and How, J. "Three dimensional receding horizon control for UAVs", AIAA Guidance, Navigation, and Control Conference and Exhibit
[3] Richards, A., and How, J.P., "Model predictive control of vehicles maneuvers with guaranteed completion time and robust feasibility, *Proceedings of 2003 American Control Conference*, 4-6 June 2003, Vol5 IEEE Denver CO, pp. 4034-40.
[4] Frey, E., Receding horizon control using random search for UAV navigation with passive, non-cooperative sensing", *AIAA, Guidance, Navigation, and Control Conference and Exhibit*, 15-18 August 2005
[5] Prazenica, R., Kurdila, A., Sharpley,R., and Evers J., "Multiresolution and adaptive path planning for maneuver of MAV in urban environment" *AIAA Guidance, Navigation, and Control Conference and Exhibit* 15-18 August 2005, San Francisco CA.
[6] Hernandez, M.L., "Optimal sensor trajectories in bearing-only tracking" *Proceedings of the Seventh International Conference on Information Fusion", FUSION 2004*, Jun-July 2004, Vol.2
[7] Frey, E. and Rock, S.M., "Trajectory generation of monocular-vision based tracking of a constant-velocity target", *Proceedings of the 2003 IEEE International Conference for Robotics and Automation, 2003*.
[8] Rathburn, D., Kragelund, S., and Pongpunwattana, A., "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environment" *Air Traffic Management for Commercial and Military Systems,* Oct 27-31 2002, Vol.2, IEEE.
[9] Anderson, M., Lawrence, W., Lopez, J., "Supplementing gradient search with genetic algorithm for air data estimation system", AIAA-1994-1931, *AIAA Applied Aerodynamics Conference,* 12th Colorado Springs, CO, June 20-222, 1994, PP. 880-889.
[10] Holland, J., "Adaptation in Natural and Artificial Systems". MIT Press, 1975.
[11] Goldberg, D. E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Publishing Company, 1989.
[12] Sushil J., and Rawlins, G., "Pareto optimality, GA-easiness and deception", *Proceedings of the Fifth International Conference on Genetic Algorithms,* pp. 118-123, Morgan Kauffmann, San Mateo, CA, 1993.
[13] Schaffer, J. D., "Some experiments in machine learning using vector evaluated genetic algorithms", Doctoral Dissertation, Vanderbilt University, 1984.
[14] Schaffer, J.D., & Grefenstette, J. J., "Multi-objective learning via genetic algorithms", *Proceedings of the 9th International Joint Conference on Artificial Intelligence,* vol. 1, pp. 593-595, 1985.
[15] Schaffer, J., D., & Morishima, A., "An Adaptive Crossover Distribution Mechanism for Genetic Algorithms", Genetic Algorithms and their Applications: *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 36-40, 1987.